Минобрнауки России

Санкт-Петербургский государственный политехнический университет

Институт информационных технологий и управления

**Кафедра «Информационные и управляющие системы»**

**КУРСОВОЙ ПРОЕКТ**

по дисциплине «Системы программирования»

**< Общее название >**

**Построение компилятора с Ассемблера.**

Выполнили

студенты гр.5084/12 А.А.Лукашин

К.С.Шубин

Руководитель

доцент В.Я.Расторгуев

«\_\_» \_\_\_\_\_\_\_\_\_\_ 2013 г.

Санкт-Петербург

2013

Оглавление

[Задание 3](#_Toc356971154)

[План работы 4](#_Toc356971155)

[Расширение таблицы команд 4](#_Toc356971156)

[Модификация кода компилятора 5](#_Toc356971157)

[Выводы 11](#_Toc356971158)

# Задание

*Вариант №7:*

Код на языке PL1

EX07: PROC OPTIONS (MAIN);

DCL A BIT (3) INIT ( 10B );

DCL B BIT (3) INIT ( 101B );

DCL C BIT (16);

C = SUBSTR((B !! A),2,3);

END EX07;

В результате первой части курсовой работы был получен эквивалент программы на языке Ассемблер.

EX07 START 0

BALR RBASE,0

USING \*,RBASE

LH 3,B

LH 4,A

SRL 4,3

OR 3,4

SLL 3,2

LH 4,TMP

NR 3,4

SRL 3,16

STH 3,C

BCR 15,RVIX

A DC 0H

DC BL2'10'

B DC 0H

DC BL2'101'

C DC 0H

DS BL2

TMP DC 0H

DC BL2'111'

RBASE EQU 5

RVIX EQU 14

END EX07

# План работы

Необходимо доработать компилятор с Ассемблера в объектное представление, дополнив его новой функциональностью. В новую функциональность входят:

1. поддержка новых команд (LH, STH, SRL, SLL, OR, NR);
2. поддержка типа BL2.

Для решения этой задачи необходимо:

1. расширить таблицу машинных команд;
2. модифицировать обработчики команд DC и DS первого и второго просмотров, а также изменить обработчик команд RX-типа.

# Расширение таблицы команд

#define DL\_ASSTEXT 29

#define DL\_OBJTEXT 100 /\*длина об'ектн. текста \*/

#define NSYM 50 /\*размер табл.символов \*/

#define NPOP 6 /\*размер табл.псевдоопер. \*/

#define NOP 12 /\*размер табл.операций \*/

Таблица машинных операций имеет следующий вид (добавленные фрагменты выделены цветом):

T\_MOP [NOP] =

{

{{'B','A','L','R',' '} , '\x05' , 2 , FRR} ,

{{'B','C','R',' ',' '} , '\x07' , 2 , FRR} ,

{{'S','T',' ',' ',' '} , '\x50' , 4 , FRX} ,

{{'L',' ',' ',' ',' '} , '\x58' , 4 , FRX} ,

{{'A',' ',' ',' ',' '} , '\x5A' , 4 , FRX} ,

{{'S',' ',' ',' ',' '} , '\x5B' , 4 , FRX} ,

{{'L','H',' ',' ',' '} , '\x48' , 4 , FRX} ,

{{'S','R','L',' ',' '} , '\x01' , 4 , FRX} ,

{{'S','L','L',' ',' '} , '\x02' , 4 , FRX} ,

{{'O','R',' ',' ',' '} , '\x16' , 4 , FRX} ,

{{'N','R',' ',' ',' '} , '\x14' , 4 , FRX} ,

{{'S','T','H',' ',' '} , '\x40' , 4 , FRX} ,

};

Также в начале второго просмотра были установлены указатели на программные обработчики новых команд:

CONT3:

T\_MOP[0].BXPROG = SRR;

T\_MOP[1].BXPROG = SRR;

T\_MOP[2].BXPROG = SRX;

T\_MOP[3].BXPROG = SRX;

T\_MOP[4].BXPROG = SRX;

T\_MOP[5].BXPROG = SRX;

T\_MOP[6].BXPROG = SRX;

T\_MOP[7].BXPROG = SRX;

T\_MOP[8].BXPROG = SRX;

T\_MOP[9].BXPROG = SRX;

T\_MOP[10].BXPROG = SRX;

# Модификация кода компилятора

***Первый просмотр:***

int FDC() /\*подпр.обр.пс.опер.DC \*/

{

if (PRNMET == 'Y') /\*если псевдооп.DC помеч.,\*/

{ /\*то: \*/

if (TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == 'F') /\* если псевдооперация DC \*/

/\* определяет константу \*/

/\* типа F, то выполнить \*/

/\* следующее: \*/

{

T\_SYM[ITSYM].DLSYM = 4; /\* уст.длину симв. = 4, \*/

T\_SYM[ITSYM].PRPER = 'R'; /\* а,призн.перемест.='R' \*/

if (CHADR % 4) /\* и, если CHADR не указ.\*/

{ /\* на границу слова, то: \*/

CHADR = (CHADR / 4 + 1) \* 4; /\* уст.CHADR на гр.сл. и\*/

T\_SYM[ITSYM].ZNSYM = CHADR; /\* запомн. в табл.симв. \*/

}

PRNMET = 'N'; /\* занулить PRNMET зн.'N'\*/

} else if (TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == 'H') {

T\_SYM[ITSYM].DLSYM = 2; /\* уст.длину симв. = 2, \*/

T\_SYM[ITSYM].PRPER = 'R'; /\* а,призн.перемест.='R' \*/

if (CHADR % 2) /\* и, если CHADR не указ.\*/

{ /\* на границу слова, то: \*/

CHADR = (CHADR / 2 + 1) \* 2; /\* уст.CHADR на гр.сл. и\*/

T\_SYM[ITSYM].ZNSYM = CHADR; /\* запомн. в табл.симв. \*/

}

PRNMET = 'N';

} else if (TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == 'B'

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[1] == 'L'

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[2] == '2')

{

T\_SYM[ITSYM].DLSYM = 2; /\* уст.длину симв. = 4, \*/

T\_SYM[ITSYM].PRPER = 'R'; /\* а,призн.перемест.='R' \*/

if (CHADR % 2) /\* и, если CHADR не указ.\*/

{ /\* на границу слова, то: \*/

CHADR = (CHADR / 2 + 1) \* 2; /\* уст.CHADR на гр.сл. и\*/

T\_SYM[ITSYM].ZNSYM = CHADR; /\* запомн. в табл.симв. \*/

} else {

CHADR += 2;

}

PRNMET = 'N';

} else

return (1); /\* иначе выход по ошибке \*/

return 0;

} else { /\*если же псевдооп.непомеч\*/

if (TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == 'B'

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[1] == 'L'

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[2] == '2')

{

T\_SYM[ITSYM].DLSYM = 2; /\* уст.длину симв. = 4, \*/

T\_SYM[ITSYM].PRPER = 'R'; /\* а,призн.перемест.='R' \*/

if (CHADR % 2) /\* и, если CHADR не указ.\*/

{ /\* на границу слова, то: \*/

CHADR = (CHADR / 2 + 1) \* 2; /\* уст.CHADR на гр.сл. и\*/

T\_SYM[ITSYM].ZNSYM = CHADR; /\* запомн. в табл.симв. \*/

} else {

CHADR += 2;

}

PRNMET = 'N';

}

return (0);

} /\*успешно завершить подпр.\*/

}

int FDS() /\*подпр.обр.пс.опер.DS \*/

{

if (PRNMET == 'Y') /\*если псевдооп.DC помеч.,\*/

{ /\*то: \*/

if /\* если псевдооперация DC \*/

( /\* определяет константу \*/

TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == 'F' /\* типа F, то

выполнить \*/

) /\* следующее: \*/

{

T\_SYM[ITSYM].DLSYM = 4; /\* уст.длину симв. = 4, \*/

T\_SYM[ITSYM].PRPER = 'R'; /\* а,призн.перемест.='R' \*/

if (CHADR % 4) /\* и, если CHADR не указ.\*/

{ /\* на границу слова, то: \*/

CHADR = (CHADR / 4 + 1) \* 4; /\* уст.CHADR на гр.сл. и\*/

T\_SYM[ITSYM].ZNSYM = CHADR; /\* запомн. в табл.симв. \*/

}

CHADR += 4;

PRNMET = 'N'; /\* занулить PRNMET зн.'N'\*/

} else if (TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == 'B'

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[1] == 'L'

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[2] == '2')

{

T\_SYM[ITSYM].DLSYM = 2; /\* уст.длину симв. = 4, \*/

T\_SYM[ITSYM].PRPER = 'R'; /\* а,призн.перемест.='R' \*/

if (CHADR % 2) /\* и, если CHADR не указ.\*/

{ /\* на границу слова, то: \*/

CHADR = (CHADR / 2 + 1) \* 2; /\* уст.CHADR на гр.сл. и\*/

} else {

CHADR += 2;

}

PRNMET = 'N';

} else if (TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == '0'

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[1] == 'H')

{

T\_SYM[ITSYM].DLSYM = 2; /\* уст.длину симв. = 2, \*/

T\_SYM[ITSYM].PRPER = 'R'; /\* а,призн.перемест.='R' \*/

if (CHADR % 2) /\* и, если CHADR не указ.\*/

{ /\* на границу слова, то: \*/

CHADR = (CHADR / 2 + 1) \* 2; /\* уст.CHADR на гр.сл. и\*/

T\_SYM[ITSYM].ZNSYM = CHADR; /\* запомн. в табл.симв. \*/

}

PRNMET = 'N';

} else

return (1); /\* иначе выход по ошибке \*/

return 0;

} else { /\*если же псевдооп.непомеч\*/

if (TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == 'B'

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[1] == 'L'

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[2] == '2')

{

T\_SYM[ITSYM].DLSYM = 2; /\* уст.длину симв. = 4, \*/

T\_SYM[ITSYM].PRPER = 'R'; /\* а,призн.перемест.='R' \*/

if (CHADR % 2) /\* и, если CHADR не указ.\*/

{ /\* на границу слова, то: \*/

CHADR = (CHADR / 2 + 1) \* 2; /\* уст.CHADR на гр.сл. и\*/

} else {

CHADR += 2;

}

PRNMET = 'N';

} else if (TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == '0'

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[1] == 'H')

{

T\_SYM[ITSYM].DLSYM = 2; /\* уст.длину симв. = 2, \*/

T\_SYM[ITSYM].PRPER = 'R'; /\* а,призн.перемест.='R' \*/

if (CHADR % 2) /\* и, если CHADR не указ.\*/

{ /\* на границу слова, то: \*/

CHADR = (CHADR / 2 + 1) \* 2; /\* уст.CHADR на гр.сл. и\*/

T\_SYM[ITSYM].ZNSYM = CHADR; /\* запомн. в табл.симв. \*/

}

PRNMET = 'N';

} else

return (1);

}

return (0);

}

***Второй просмотр:***

int SDC() /\*подпр.обр.пс.опер.DC \*/

{

char \*RAB; /\*рабочая переменная \*/

RX.OP\_RX.OP = 0; /\*занулим два старших \*/

RX.OP\_RX.R1X2 = 0; /\*байта RX.OP\_RX \*/

if (!memcmp(TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND, "F'", 2))

/\* если операнд начинается\*/

/\* с комбинации \*/

/\* F', \*/

/\* то \*/

{

RAB = strtok /\*в перем. c указат.RAB \*/

( /\*выбираем первую лексему \*/

(char\*) TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND + 2, /\*операнда

текущей карты \*/

"'" /\*исх.текста АССЕМБЛЕРА \*/

);

RX.OP\_RX.B2D2 = atoi(RAB); /\*перевод ASCII-> int \*/

RAB = (char \*) &RX.OP\_RX.B2D2; /\*приведение к соглашениям\*/

swab(RAB, RAB, 2); /\* ЕС ЭВМ \*/

STXT(4);

} else /\*иначе \*/

if (!memcmp(TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND, "H'", 2)) /\* если

операнд начинается\*/

/\* с комбинации \*/

/\* H', \*/

/\* то \*/

{

RAB = strtok /\*в перем. c указат.RAB \*/

( /\*выбираем первую лексему \*/

(char\*) TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND + 2, /\*операнда

текущей карты \*/

"'" /\*исх.текста АССЕМБЛЕРА \*/

);

RR.OP\_RR.OP = 0;

RR.OP\_RR.R1R2 = atoi(RAB); /\*перевод ASCII-> int \*/

RAB = (char \*) &RR.OP\_RR.R1R2; /\*приведение к соглашениям\*/

STXT(2);

}

else if (!memcmp(TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND, "BL2'", 4))

/\* если операнд начинается\*/

/\* с комбинации \*/

/\* BL2', \*/

/\* то \*/

{

RAB = strtok /\*в перем. c указат.RAB \*/

( /\*выбираем первую лексему \*/

(char\*) TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND + 4, /\*операнда

текущей карты \*/

"'" /\*исх.текста АССЕМБЛЕРА \*/

);

RAB = strtok /\*в перем. c указат.RAB \*/

( /\*выбираем первую лексему \*/

(char\*) TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND + 4, /\*операнда

текущей карты \*/

"'" /\*исх.текста АССЕМБЛЕРА \*/

);

int value = atoi(RAB); /\* получение числового значения лексемы \*/

int len = strlen(RAB); /\* получение длины лексемы \*/

value <<= (16 - len); /\* сдвиг битов к началу поля данных \*/

RAB = (char \*) &value;

swab(RAB, RAB, 2); /\* приведение к соглашениям ЕС ЭВМ \*/

char buf[2] = /\* получение размера резервируемой памяти \*/

{ TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[2], '\x0' };

int bytes = atoi(buf);

RR.OP\_RR.OP = 0; /\*занулим два старших \*/

RR.OP\_RR.R1R2 = 0; /\*занулим RX.OP\_RX.B2D2 \*/

memcpy(RR.BUF\_OP\_RR, &value, 2);

STXT(2);

} else

return (1); /\*сообщение об ошибке \*/

/\*формирование TXT-карты \*/

return (0); /\*успешн.завершение подпр.\*/

}

int SDS() /\*подпр.обр.пс.опер.DS \*/

{

if ( /\* если операнд начинается\*/

TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == 'F' /\* с комбинации F \*/

) /\* то: \*/

{

RX.OP\_RX.OP = 0; /\*занулим два старших \*/

RX.OP\_RX.R1X2 = 0; /\*байта RX.OP\_RX \*/

RX.OP\_RX.B2D2 = 0; /\*занулим RX.OP\_RX.B2D2 \*/

STXT(4); /\*формирование TXT-карты \*/

} else if (TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == 'B' /\* если

операнд начинается\*/

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[1] == 'L' /\* с комбинации BL2 \*/

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[2] == '2') {

RR.OP\_RR.OP = 0; /\*занулим два старших \*/

RR.OP\_RR.R1R2 = 0;

STXT(2); /\*формирование TXT-карты \*/

} else if ( /\*иначе \*/

TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[0] == '0' /\* если операнд начинается \*/

&& TEK\_ISX\_KARTA.STRUCT\_BUFCARD.OPERAND[1] == 'H' /\* с комбинации 0H \*/

) {

if (CHADR % 2) /\* выставляем счетчик адреса \*/

{

CHADR = (CHADR / 2 + 1) \* 2;

}

} else

return (1); /\*сообщение об ошибке \*/

return (0);

}

# Выводы

В рамках второго этапа курсовой работы по написанию компилятора с языка ассемблер были выполнены все поставленные задачи:

1. Расширена таблица машинных команд
2. Модифицированы обработчики команд DC и DS
3. Существующий компилятор доработан с учетом новых правил

В результате выполнения работы был получен объектный модуль. Проверить правильность его работы можно с помощью абсолютного загрузчика и эмулятора машины.